Abstract

Edge detection is an important task in image processing. It is used as a preprocessing step in many enhancement and image understanding applications. Although the concepts are simple to understand, it is a very challenging technique to perfect. Various types of filtering are presented and the Marr-Hildreth and Canny edge detectors are constructed and discussed.
Edge Detection

Edge detection is an important technique in image processing. It has been studied intensively, and a large body of literature exists.

Applications of edge detection include:

- Image segmentation
- Feature Identification

Edge detection algorithms use combinations of filtering, thresholding and logic.

Popular algorithms: Marr-Hildreth and Canny.
Edge Detection by Gradient Threshold

The image gradient, as computed by the Sobel gradient operator or other means, highlights points on edges.

Grayscale Image

Gradient Image
Edge Detection by Gradient Threshold

Edges are detected by thresholding the gradient image. The question is “Where to set the threshold?”

Grayscale Histogram

Gradient Histogram

Automatic threshold setting is easier for grayscale images than for gradient images, where there is typically only one maximum.
Edge Detection by Gradient Threshold

Threshold = 32

Threshold = 64

Threshold = 96

Results are sensitive to threshold setting.

How can we create an automatic edge detection algorithm?
Marr-Hildreth Edge Detector


The Marr-Hildreth algorithm for edge detection is based on the zero-crossings of the Laplacian of the Gaussian operator. The Gaussian operator smoothes the image and the Laplacian operator computes the second derivative. Edges are found at zero-crossings of the resulting image.

Ellen Hildreth
Log of Gaussian (LoG) Filter

**FIGURE 10.14**
Laplacian of a Gaussian (LoG).
(a) 3-D plot.
(b) Image (black is negative, gray is the zero plane, and white is positive).
(c) Cross section showing zero crossings.
(d) 5 × 5 mask approximation to the shape of (a).
Marr-Hildrith Edge Detector

GW Figure 10.15: (a) Original (b) Sobel Filtered (shown for comparison) (c) Gaussian LPF (d) Laplacian (e) LoG of original (f) Thresholded (g) Zero-crossings
Marr-Hildreth Example

;Value of sigma
s=4

;Get the wheel image
fname='d:\harvey\classes\simg782\2004\lectures\lecture_15\'
A=read_image(fname+'wheel.gif')

;Set up 15x15 LoG filter with sigma=1
h=LoG(15,s)

;Filter the wheel image
A1=convol(float(A),h)

;Threshold the LoG image
A2=bytscl(A1 GT 0)

;HPF to highlight the edges
hp=[[[-1,-1,-1],[-1,8,-1],[-1,-1,-1]]
A3=bytscl(convol(A2,hp) GT 0)

window,/free,xs=512,ys=512
TVscl,A,0,0
TVscl,A1,256,0
TV,A2,0,256
TV,A3,256,256
str=string(format='("Sigma = ",I1)',s)
xyouts,256,20,str,alignment=0.5,charsize=2,charthick=2,color=255,/device
Marr-Hildreth Example (cont)

Edge detection with $\sigma = 3$.

Lower left – original image

Lower right – after LoG

Upper left – after threshold

Upper right – detected zero-crossings
Marr-Hildreth Example (cont)

\[
\begin{array}{|c|c|}
\hline
\sigma &= 1 & \sigma &= 2 \\
\sigma &= 3 & \sigma &= 4 \\
\hline
\end{array}
\]
Canny Edge Detection

We will use the Canny edge detection algorithm as an example of the use a number of techniques in combination to detect and refine edge decisions.

One of the most successful edge detection systems is the Canny Edge Detector

What is an Edge?

An edge is a place where there is a rapid change in the brightness (or other property) of an image.

The brightness along a horizontal slice through the image is plotted.

We see that there are steps where the brightness changes significantly.

How can this insight be used in the construction of an edge detector?

What problems would be expected?
Edge Model

(a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.
Gradient

The gradient is a good tool for the detection of rapid changes.

The image at the right has been processed by the SOBEL operator, which detects gradients.

The spikes represent locations of edges.

One problem is sensitivity to noise.
Effect of Noise

Gonzalez and Woods, Figure 10.7.

First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and $\sigma = 0.0, 0.1, 1.0$ and $10.0$, respectively.

Second column: first derivative images and gray-level profiles.

Third column: second-derivative images and gray-level profiles.
Smoothing Filter

Many different kinds of smoothing filters can be used.

A popular choice is the gaussian filter.

\[ h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \]

\[ = \left( \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \right) \left( \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \right) \]

The filter is *separable* and *scalable*. The filter is rotationally symmetric and its transform \( H(u, v) \) is also gaussian.
Gaussian Filter

A gaussian filter can be constructed to approximate the 2D gaussian surface.

It is useful to keep the size of the filter array small while still accomplishing the desired smoothing.

A $5 \times 5$ gaussian filter with $\sigma = 0.8$ is

$$h = \frac{1}{198} \begin{bmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 10 & 23 & 10 & 1 \\ 2 & 23 & 50 & 23 & 2 \\ 1 & 10 & 23 & 10 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{bmatrix}$$
Smoothing + Gradient

Gradient without filter

Gradient with filter
Another Example

Original Image

Smoothed with $5 \times 5$ gaussian

Original + Gradient

Filtered + Gradient
Gradient Profile

A closeup of the gradient at an edge is shown in the surface plots below. A considerable amount of smoothing of the original image may be required to reveal the true edges in the gradient image.

Gradient without smooting  Smooted with $5 \times 5$  Smooted with $15 \times 15$

$\sigma = 0.8$ before gradient  $\sigma = 2$ before gradient

Increasing the amount of smoothing reduces the edge noise. We still need a way to decide whether a pixel is on an edge or not.

A simple threshold technique would produce noisy fat edges.
Edges by Threshold

When edges are located by a global threshold on the smoothed gradient image the result is not satisfactory. Some edges are missed and others are too broad.
Edge Decision

A global threshold does not have the flexibility to produce acceptable results.

- The tops of some ridges in the gradient image are too low and the edges are missed.

- Some ridges are far above the threshold. This produces fat edge profiles.

The Canny algorithm uses a combination of tests to identify pixels on edges.

If a pixel in the gradient image is on the top of a ridge it must have a larger value than its neighbors along a path orthogonal to the ridge line.

This rule can be used to construct a test called “non-maximal suppression.” This will tend to thin the edges.

Non-maximal suppression is followed by a alternating series of threshold tests to adapt to changing elevation.
Canny Algorithm

FUNCTION canny,$
  image,$ ;Input image array (required)
  sigma,$ ;Smoothing filter spread (required)
  tlow,$ ;Low threshold (required)
  thigh,$ ;High threshold (required)
  SMOOTHED_IMAGE=smoothedim,$ ;Smoothed image
  GRAD_X=delta_x,$ ;X gradient
  GRAD_Y=delta_y,$ ;Y gradient
  GRAD_M=magnitude,$ ;Gradient magnitudes
  GRAD_A=dir_radians,$ ;Gradient angles
  NMS=nms ;Internal possible edge map
Canny Algorithm

imsize=Size(image)
IF imsize[0] NE 2 THEN Message,’Image must be 2D,/Error’
;Perform Gaussian smoothing
gaussian_smooth,image, rows, cols, sigma, smoothedim
;Compute X and Y gradients
derivative_x_y,smoothedim, rows, cols, delta_x, delta_y
;Compute magnitude of gradient
magnitude=Sqrt(Float(delta_x)^2 + Float(delta_y)^2)
;Compute direction of gradient (for output)
radian_direction,delta_x, delta_y, dir_radians, -1, -1

;Perform non-maximal suppression
Non_max_supp,magnitude, delta_x, delta_y, rows, cols, nms
;Apply hysteresis to mark edge pixels
Apply_Hysteresis,magnitude, nms, rows, cols, tlow, thigh, edge
Return,BytScl(edge)
END
Canny Results $\sigma = 1.5 \ t_{low} = 0.5$

$t_{high} = 0.7$

$t_{high} = 0.8$

$t_{high} = 0.9$

$t_{high} = 0.8$
Canny Results $\sigma = 1.5$ $t_{low} = 0.5$

$t_{high} = 0.7$

$t_{high} = 0.8$

$t_{high} = 0.9$

$t_{high} = 0.8$
Canny Edge Detector – Wheel

Original

\( \sigma = 2 \) \( t_{low} = 0.3, \ t_{high} = 0.6 \)

\( \sigma = 3 \) \( t_{low} = 0.3, \ t_{high} = 0.6 \)

\( \sigma = 1 \) \( t_{low} = 0.3, \ t_{high} = 0.6 \)
Canny Detector

Original

\[ \sigma = 2 \quad t_{low} = 0.3, \quad t_{high} = 0.6 \]
WASP LWIR Image

Original

\[ \sigma = 1 \quad t_{low} = 0.3, \quad t_{high} = 0.5 \]